Parsing as Deduction

Joseph Kühner

March 24, 2007
Parsing algorithms for various types of languages are represented in a formal logic framework as deduction systems, where items (formulas) describe the grammatical status of strings, and inference rules produce new items from already generated items. On this more abstract level, Parsing Deduction Systems reflect the structure of parsers in a clear and concise manner and provide unified tools for the proof of correctness, completeness and complexity analysis.
Parsing Deduction System

Parsing of CFG - Example CYK
  CYK Parsing Algorithm
  CYK Deductive Parsing System

Tree Adjoining Grammars

Parsing Deduction for Tree Adjoining Grammars (TAG)

Agenda-Chart Deduction Procedure
A parsing deduction system can be specified as

- A set of items
- A set of axioms
- A set of inference rules
- A subclass of items, the goal items

The general form of a rule of inference is

$$\frac{A_1 \ldots A_k}{B} \langle \text{side conditions on } A_1, \ldots A_k, B \rangle$$

The antecedents $A_1, \ldots A_k$ and the consequent $B$ of the rule are items. Axioms can be represented as inference rules with empty set of antecedents.
A derivation of an item $B$ from assumptions $A_1, \ldots, A_m$ is a sequence of items $S_1, \ldots, S_n$ where $S_n = B$ and $S_i$ is either an axiom or there is a rule $R$ and items $S_{i_1}, \ldots, S_{i_k}$ with $i_1, \ldots, i_k < i$ such that:

\[
\frac{S_{i_1} \ldots S_{i_k}}{S_i} \quad \langle \text{side conditions} \rangle
\]

We write $A_1, \ldots, A_m \vdash B$. 
Let $G = (N, \Sigma, P, S)$ be a CFG in CNF, $w = w_1 \ldots w_n$ a string in $\Sigma$. Compute sets $T_{ij}$, $1 \leq i \leq j \leq n$, of nonterminals such that $A \in N$ belongs to $T_{ij}$ iff $A^* \rightarrow w_i \ldots w_j$.

For $1 \leq i \leq j \leq n$ set $T_{ij} = \emptyset$.

- For $1 \leq i \leq n$ add nonterminal $A$ to $T_{ii}$ iff $A \rightarrow w_i$
- For $1 \leq i < j \leq n$ add nonterminal $A$ to $T_{ij}$ iff there is a rule $A \rightarrow BC$ and $k \in \{1, \ldots, j - 1\}$ with $B \in T_{ik}$ and $C \in T_{k+1,j}$
- $w \in L(G)$ iff $S \in T_{1n}$
Let $G = (N, \Sigma, P, S)$ be a CFG in CNF, $w = w_1 \ldots w_n$ a string in $\Sigma^*$. Consider items (formulas) $[A, i, j]$, $A \in N$, $1 \leq i \leq j \leq n$, which state that $A \rightarrow^* w_i \ldots w_j$.

- Item form: $[A, i, j]$
- Axioms: $[A, i, i]$ \{ $A \rightarrow w_i$ \}
- Goals: $[S, 1, n]$
- Inference Rules: $[B, i, k] [C, k + 1, j] \leftarrow [A, i, j]$ \{ $A \rightarrow BC$ \}
Lemma

If an item \([A, i, j]\) can be derived in the deduction system then \(A \rightarrow^* w_1 \ldots w_j\) in the grammar \(G\).
Proof.
We prove the lemma by induction on $l = j - i$.

If the item $[A, i, i]$ can be derived, it is an axiom; this means that $A \rightarrow w_i$ is a production in $G$.

If $l > 0$ and the item $[A, i, j]$ can be derived then an inference rule must have been applied. This means that there exist a production $A \rightarrow BC$ in $G$ and $1 \leq k \leq j - 1$ and items $[B, i, k]$ and $[C, k + 1, j]$, both derivable, which infer $[A, i, j]$. By induction $B \xrightarrow{*} w_i \ldots w_k$ and $C \xrightarrow{*} w_{k+1} \ldots w_j$. Applying the production $A \rightarrow BC$ one finds that $A \xrightarrow{*} w_i \ldots w_j$. 

\[\square\]
Correctness

Theorem

*If the item \([S, 1, n]\) is derivable in the deduction system then the string \(w_1 \ldots w_n\) belongs to \(L(G)\).*

Proof.

By the lemma, if \([S, 1, n]\) is derivable, we have \(S \rightarrow^* w_1 \ldots w_n\).

Hence \(w_1 \ldots w_n \in L(G)\).
Completeness

Theorem
If \( w = w_1 \ldots w_n \in L(G) \) then item \([S, 1, n]\) can be derived in the deduction system
A tree adjoining grammar (TAG) is a quintuple $G = (N, \Sigma, S, I, A)$ where

- $N$ is a set of nonterminals
- $\Sigma$ a set of terminals
- $S$ a distinguished nonterminal, the start symbol
- $I$ a set of initial trees
- $A$ a set of auxiliary trees

The trees in $I \cup A$ are called elementary.
Initial Tree, Auxiliary Tree

\[ \alpha = \begin{array}{c} A \\ w_1 \ldots \ w_n \end{array} \quad \beta = \begin{array}{c} B \\ w_1 \ldots w_j B w_{k+1} \ldots w_n \end{array} \]

**Figure:** Initial tree \( \alpha \)  Auxiliary tree \( \beta \)
Adjunction of tree $\beta$ at node $\nu$ in tree $\alpha$

Given
- A tree $\alpha$ with an inner node $\nu$ labelled $B$
- An auxiliary tree $\beta$ with root and foot node labelled $B$.

Adjoin
- Excise subtree of $\alpha$ rooted at $\nu$
- Insert $\beta$ at $\nu$
- Append previously excised subtree at foot node of $\beta$
Trees before Adjunction

$\alpha = \begin{array}{c}
A \\
B \\
\end{array}
\quad \beta = \begin{array}{c}
B \\
\end{array}$

Figure: Root and foot node of the auxiliary tree $\beta$ are labelled $B$. $\beta$ can be adjoint to tree $\alpha$ at node $\nu$ labelled $B$. 
Tree after Adjunction

**Figure:** Tree $\gamma$ results from adjoining $\beta$ to $\alpha$ at node $\nu$ labelled $B$. 
Derivable Trees

Adjoin trees $\beta_1, \ldots, \beta_k$ at distinct addresses $a_1, \ldots, a_k$ in $\alpha$:

$\alpha[\beta_1 \rightarrow a_1, \ldots, \beta_k \rightarrow a_k]$

The set $D(\mathcal{G})$ of derivable trees is the smallest set such that

$\triangleright I \cup A \subseteq D(\mathcal{G})$

$\triangleright$ For all $\alpha \in I \cup A$, the set $D(\alpha, \mathcal{G})$ of trees $\alpha[\beta_1 \rightarrow a_1, \ldots, \beta_k \rightarrow a_k]$ where $\beta_1, \ldots, \beta_k \in D(\mathcal{G})$, is a subset of $D(\mathcal{G})$

Valid derivations in $\mathcal{G}$

$\triangleright$ Trees in $D(\alpha_S, \mathcal{G})$ where $\alpha_S \in I$ with root is labelled with start symbol $S$. 
Items $[ν^*, i, j, k, l]$ resp. $[ν_*, i, j, k, l]$, where

- $ν$ is a node in an elementary tree $α$
- $0 \leq i \leq l \leq n$ are string positions
- $j$ and $k$ undefined or instantiated to positions $i \leq j \leq k \leq l$.
- Dot position keeps track of aduction at node $ν$
Invariants

Item $[\alpha@a^\bullet, i, j, k, l]$ specifies

- There is a tree $\tau \in D(\alpha|a)$ such that the frontier of $\tau$ is $w_{i+1} \ldots w_j \text{Label}(\alpha) w_{k+1} \ldots w_l$
- Adjunction at node $\alpha@a$ may involve in derivation of $\tau$.

Item $[\alpha@a_\bullet, i, j, k, l]$ specifies

- There is a tree $\tau \in D(\alpha|a)$ such that the frontier of $\tau$ is $w_{i+1} \ldots w_j \text{Label}(\alpha) w_{k+1} \ldots w_l$
- Adjunction at node $\alpha@a$ must not involve in derivation of $\tau$. 
\[ \alpha = \nu^\bullet \]

Figure: Tree \( \alpha \) illustrates item \([\nu^\bullet, i, j, k, l]\).
Invariant

Items $[\alpha @ a^\bullet, i, -, -, l]$ and $[\alpha @ a^\bullet, i, -, -, l]$ specify similar invariants except that there is no foot node in the frontier of $\tau$. 
Parsing Deduction System for TAG

Item Form: \[
\begin{align*}
\left[ \nu^\bullet, i, j, k, l \right] \\
\left[ \nu, i, j, k, l \right]
\end{align*}
\]

Terminal Axiom: \[
\left[ \nu^\bullet, i, _, _, i + 1 \right] \quad \text{Label}(\nu) = w_{i+1}
\]

\(\epsilon\) Axiom: \[
\left[ \nu^\bullet, i, _, _, i \right] \quad \text{Label}(\nu) = \epsilon
\]

Foot Axiom: \[
\left[ \beta @ \text{Foot}(\beta), j, j, k, k \right] \quad \beta \in A
\]

Goals: \[
\left[ \alpha @ \epsilon^\bullet, 0, _, _, n \right] \quad \alpha \in I, \text{ Label}(\alpha @ \epsilon) = S
\]
Inference Rules:

Complete Unary: \[
[\alpha @ a_1 \circ, i, j, k, l] \\
[\alpha @ a_\bullet, i, j, k, l]
\]

Complete Binary: \[
[\alpha @ a_1 \circ, i, j, k, l] \\
[\alpha @ a_2 \circ, l, j', k', m] \\
[\alpha @ a_\bullet, i, j \cup j', k \cup k', m]
\]

No Adjoin: \[
[\nu \circ, i, j, k, l] \\
[\nu \bullet, i, j, k, l]
\]

Adjoin: \[
[\beta @ \epsilon \circ, i, p, q, l] \\
[\nu \circ, p, j, k, q] \\
[\nu \bullet, i, j, k, l]
\] \quad \beta \in \text{Adj}(\nu)
Binary Completion

\[ \alpha = \]

\[ w_{i+1} \ldots w_j A w_{k+1} \ldots w_l w_{l+1} \ldots \ldots w_m \]

**Figure:** Tree \( \alpha \) Illustrates Binary Completion.
Correctness

Lemma
Let $[\nu^\bullet, i, j, k, l]$ (resp. $[\nu^\bullet, i, j, k, l]$) be a derivable item in the above specified deduction system, then there is an elementary tree $\alpha$ with inner node $\nu^\bullet$ (resp.$\nu^\bullet$) and a derived tree $\tau$ in $D(\nu, G)$ whose frontier string is equal to $w_{i+1} \ldots w_j \text{Label}(\alpha) w_{k+1} \ldots w_l$. 
Correctness Case Adjunction

Item \([\nu^\bullet, i, j, k, l]\) and is generated by the adjunction rule

\[
\begin{array}{c}
[\beta \otimes \epsilon^\bullet, i, p, q, l] [\nu^\bullet, p, j, k, q] \\
\downarrow \\
[\nu^\bullet, i, j, k, l]
\end{array}
\]

Induction hypothesis can be applied to both antecedents.

- There is a tree \(\tau \in D(\nu, G)\) with frontier 
  \(w_{p+1} \ldots w_j \text{Label}(\alpha) w_{k+1} \ldots w_q\)

- a tree \(\beta' \in D(\beta, (G)\) which with frontier 
  \(w_{i+1} \ldots w_p \text{Label}(\beta) w_{q+1} \ldots w_l\).

- Adjoin \(\beta'\) to \(\alpha\) at node \(\nu\) to obtain a tree with frontier 
  \(w_{i+1} \ldots w_j \text{Label}(\alpha) w_{k+1} \ldots w_l\)
Corollary

If the goal item \([\alpha \bullet \epsilon, 0, _, _, n]\), where \(\alpha \in I\), \(\text{Label}(\alpha \bullet \epsilon) = S\), can be derived in the deduction system, then the string \(w_1 \ldots w_n\) can be derived in the TAG \(G\).
Completeness

Theorem
Suppose that the string \( w = w_1 \ldots w_n \) can be derived in the TAG. Then the goal item \([\alpha^\bullet, 0, -, -, n]\) can be derived in the deduction system.
1. Initialize the chart to the empty set and the agenda to the set of axioms of the deduction system.

2. Repeat the following steps until the agenda is exhausted:
   2.1 Select an item from the agenda, called the trigger item, and remove it.
   2.2 Add the trigger item to the chart, if necessary.
   2.3 If the trigger item was added to the chart, generate all items that are new immediate consequences of the trigger item together with all the items in the chart, and add these generated items to the agenda.

3. If a goal item is in the chart, the goal is proved and the string is recognized, otherwise it is not.
Correctness

Theorem

Suppose that in the above described procedure the agenda has been initialized with items $A_1, \ldots, A_k$ and item $I$ has been placed in the chart, then $A_1, \ldots, A_k \vdash I$. 
Proof.

Induction on the stage number #$I$

- Item with #$I = n > 0$ added to the agenda by step (2.3)
- There are items $J_1, \ldots, J_m$ in the chart and a rule instance such that

  \[
  \frac{J_1 \ldots J_m}{I}
  \]

- #$J_i < n$ for each $1 \leq i \leq m$. By the induction hypothesis
- $J_i$ has a derivation $\Delta_i$ from $A_1, \ldots, A_k$.
- $\Delta_1, \ldots, \Delta_m, I$ is derivation of $I$ from $A_1, \ldots, A_k$. 

\[\Box\]
Completeness

**Theorem**

Suppose that $A_1, \ldots, A_k \vdash I$ in the parsing deduction system. Then item $I$ is in the chart at step (3).
Proof.
We show completeness by induction on the length of any derivation $D_1, \ldots, D_n$ of $I$ from $A_1, \ldots, A_k$.
If $n = 1$, we have $D_1 = I$ and $I$ is an axiom $A_i$ for some $i$. $I$ will thus be placed in the agenda at step (1) and $\#(I) = 0$. By the fairness assumption $I$ will be removed from the agenda after at most $k$ iterations of step (2). When this is done, $I$ will be added to the chart or the chart already contains the same item. \qed
Let \( n \geq 1 \) and assume the claim for derivations of length less than \( n \). Consider a derivation \( D_1, \ldots, D_n = I \) of \( I \) from \( A_1, \ldots, A_k \). Either \( I \) is an axiom, in which case we just have shown the claim, or there are indices \( i_1, \ldots, i_m < n \) such that there is an inference rule

\[
\frac{D_{i_1} \ldots D_{i_m}}{I} \quad \text{(side conditions)}
\]

with side conditions satisfied. By definition of derivation, each prefix \( D_1, \ldots, D_{i_j}, (1 \leq j \leq m) \), of \( D_1, \ldots, D_n \) is a derivation of \( D_{i_j} \) from \( A_1, \ldots, A_k \). By induction hypothesis, all items \( D_{i_j} \) are in the chart. Note \( I_p \) the item among the \( D_{i_j}' \) that was added latest to the chart. Then it will be the trigger item for the application of the above rule. Thus \( I \) will be added to the agenda. Since step (2.3) can only add a finite number of items to the agenda, item \( I \) will eventually be considered at steps (2.1) and (2.2) and added to the chart, if not already there.